



Fecha: / /

Programación III

Alumno:

Nombre: Correo:

“Aterrizar” es una empresa orientada a la venta de viajes de diferente naturaleza (terrestres, marítimos y aéreos) y requiere un sistema **modelado e implementado observando los principios de la POO** para administrar su catálogo en vigencia.

A tal efecto, se solicita proveer las siguientes clases, cada una con las características que se indican:

1) Clase CViaje, (archivo CViaje.cs), que presenta:

- Los datos miembros de instancia:
 - Código:** cadena de caracteres.
 - Origen:** cadena de caracteres.
 - Destino:** cadena de caracteres.
 - Precio:** Monto (\$) a abonar por el viaje. Número real (punto flotante de precisión **simple**).
- Un método constructor parametrizado que recibe como argumentos el código, el origen y el destino del viaje.
- Los métodos **GetOrigen()**, **GetDestino()** y **GetCodigo()**.
- Una **propiedad** de lectura y escritura **PrecioViaje**, que permita acceder a la variable de instancia asociada al precio del viaje, tanto para el establecimiento del monto (\$) como para la obtención del mismo.
- Un método **darPrecio(cuotas)** que recibiendo como argumento la cantidad de cuotas: 1, 3, 6 o 12, devuelve como número real de precisión simple, el precio a pagar, según la aplicación de un recargo de 0%, 10%, 20% y 40% respectivamente.
- Un método redefinible **darDatos()**, que sin argumento alguno, devuelve una cadena de caracteres que concatena los valores de todos los datos miembros.
- Un método que implementa el procedimiento **CompareTo(obj)** de la interfaz estándar **IComparable**. Dicho método deberá realizar la comparación en base a los datos miembro origen y destino. El método devolverá el entero cero si las dos instancias en comparación comparten el mismo origen y destino, el entero 1 si sólo comparten origen, el entero 2 si sólo comparten destino y el entero 3 si no comparten ni origen ni destino.

Instancia A		Instancia B		A.CompareTo(B)
Origen	Destino	Origen	Destino	
Salta	Trelew	Salta	Trelew	0
Salta	Trelew	Salta	Río Gallegos	1
Salta	Trelew	Córdoba	Trelew	2
Salta	Trelew	Córdoba	Río Gallegos	3
Salta	Trelew	Trelew	Salta	3

2) Clase CAereo (archivo CAereo.cs) , la cual:

- Hereda de CViaje, **reutilizando de ella todo lo que sea posible**.

E incluye:

- Una variable miembro para el impuesto a viajes aéreos, que representa un monto de dinero (\$) a adicionar sobre el precio, siendo un número real (punto flotante de precisión simple) y valor **único** para todas las instancias de la clase CAereo.
- La variable miembro para el dato tipo de vuelo (cabotaje o internacional), particular de cada instancia.
- Un método para establecer el impuesto aéreo, **incluso antes de que se haya creado instancia alguna de CAereo**.

- e) Un método constructor con los parámetros para código, origen, destino, precio y tipo de vuelo.
 - f) Un método **darDatos()**, que **redefina formalmente** el método heredado, de modo que contemple también los datos miembros agregados a la clase hija (impuesto a viajes aéreos y tipo de vuelo).
- 3) Clase CCatalogo (archivo CCatalogo.cs), que presenta:
- a) Una única **variable miembro**, de tipo colección **lista dinámica no genérica** de nombre **listado** y un constructor por defecto, que al momento de crear una instancia de CCatalogo, instancie también a **listado** con **cantidad inicial de elementos en cero**.
 - b) Un método **Registrar()** que recibiendo como argumentos código, origen, destino, precio y tipo de vuelo, verifique que para dicho **listado** no exista ya registrado un viaje de cualquier tipo con idéntico código. Si así fuera, que cree una instancia de aéreo con los datos recibidos, la agregue a **listado** y devuelva el valor lógico Verdadero. Cuando el agregado, por cualquier motivo, sea imposible de realizar, retornará el valor lógico Falso.
 - c) Un método **Remover()**, que recibe como argumento un código de viaje y elimina de la colección registrada en **listado**, el objeto cuyo código es coincidente con el argumento y retornando en tal caso, el valor lógico Verdadero. Cuando la remoción, por cualquier motivo sea imposible de realizar, retornará el valor lógico Falso.
 - d) Un método sobrecargado **Remover()**, que sin recibir argumento alguno, remueve todas las instancias de viajes de **listado**.

Criterios:

Tiempo de examen: 3 (tres) horas.

Se contemplará:

- **La entrega de los tres archivos CViaje.cs, CAereo.cs y CCatalogo.cs.**
- El cumplimiento de los tiempos y pautas de examinación.
- La prolijidad general y la claridad conceptual en el desarrollo.
- Las técnicas, metodologías y estrategias de programación aplicadas.
- La eficiencia en el empleo y la gestión de los recursos de memoria.
- La eficiencia en el empleo de estructuras de control y gestión del flujo de ejecución.
- La correcta promoción de tipos, que debe ser estricta y explícita.
- La observancia de los principios de la P.O.O. estudiados y el grado de reutilización del código.
- Importante: No se requiere validación de los datos de entrada.

V. P-20-2P-1C-TU- S.d.J.